

Tarea 2

Authors:

Eliécer Mora
Sergio Guillén

Running pi.c

pi with 100000000 steps is 3.141593 in 1.298140 seconds

This is the original file provided with no omp functionality addition.

Running pi_omp_private.c

pi with 100000000 steps is 3.141593 in 0.473014 seconds

Each master thread executing the parallel region has a private copy of the variable sum that is created by the reduction clause on the omp for statement. The master thread and all threads in its region have a private copy of the variable x that is created by the private clause on the parallel loop construct.

Running pi_omp_teams.c

pi with 100000000 steps is 0.000000 in 1.376085 seconds

The pi_omp_teams.c shows how the target, teams, and distribute constructs are used to execute a loop in a target region. The teams construct creates a league and the master thread of each team executes the teams region. The distribute construct schedules the subsequent loop iterations across the master threads of each team.

Running pi_omp_thread.c

pi with 1 Threads is 3.141588 in 1.941776 seconds

pi with 2 Threads is 3.141592 in 1.451616 seconds

pi with 3 Threads is 3.141593 in 1.408985 seconds

pi with 4 Threads is 3.141963 in 0.508163 seconds

The number of threads of a parallel region can be set by the OMP_NUM_THREADS environment variable on the parallel directive with the num_threads clause. Results seem to show that threads are not actually set to that exact number but this is maximum only. That will be the most likely reason why we couldn't get linear reduction on time.